KM1M7CF0 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro® Family

Directory Information

Document	Driver reference guide and revision history.
Library	Driver header and source files.
SampleCode	Driver sample code.

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

TABLE OF CONTENTS

1 DOCUMENT 4

2 LIBRARY 5

3 SAMPLECODE 6

4 SAMPLECODE\ISP 7

5 SAMPLECODE\STDDRIVER 8

System Setting (SYS) 8
Clock Generator (CLK) 8
General Purpose I/O (GPIO) 8
Watchdog Timer (WDT) 8
16-bit Timer (TIMER) 8
Power Supply Voltage Detection (LVD) 9
Clock Monitoring Circuit (CKM) 9
Power Control PWM (GPWM) 10
Multi Feedback Assist (MFA) 10
Digital-to-Analog Converter (DAC) 10
Comparator (CMP) 10
Analog to Digital Converter (ADC) 11
UART Interface Controller (UARTA) 11
UART Interface Controller (UARTB) 11
UART Interface Controller (UARTC) 12
Serial Peripheral Interface (SPIA) 12
Serial Peripheral Interface (SPIB) 12
Serial Peripheral Interface (SPIC) 12
I²C Serial Interface Controller (I2C) 13
SMBus Interface (SMBUS) 14
CRC Controller (CRC) 15
Flash Memory Controller (FMC) 15
DMA Controller (DMA) 15
CAN Controller (CAN) 15

Real Time Clock (RTC) **16**
LIN Controller (LIN) **16**

1 Document

CMSIS.html	Document of CMSIS version 5.1.1.
NuMicro KM1M7CF0 Series CMSIS BSP Driver Reference Guide.chm	This document describes the usage of drivers in KM1M7CF0 BSP.
NuMicro KM1M7CF0 Series CMSIS BSP Revision History.pdf	This document shows the revision history of KM1M7CF0 BSP.
NuMicro KM1M7CF0 Series CMSIS BSP Driver Revision History Details.xlsx	This document shows the revision history details of KM1M7CF0 BSP Driver.
LICENSE.txt	Document of Apache License.

2 Library

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V5.1.1 definitions by Arm® Corp.
Device	CMSIS compliant device header file.
StdDriver	All peripheral driver header and source files.
XMODEM	Specific libraries for FMC sample code

3 SampleCode

FlashOption	Sample code to create flash option data for KM1M7CF0 series MCU.
ISP	Sample codes for In-System-Programming.
StdDriver	Sample code to demonstrate the usage of KM1M7CF0 series MCU peripheral driver APIs.
Template	A project template for KM1M7CF0 series MCU.

* The following IDE versions have been confirmed to work.

IAR: Ver 9.1

KEIL: uVision 5.36

GCC: NuEclipse_V1.02.026

4 SampleCode\ISP

ISP_UART	In-System-Programming sample code through UART interface.

5 SampleCode\StdDriver

System Setting (SYS)

SYS_Reset	Reset, refer to reset factor, clear reset factor.
SYS_ReadProductID	Read and show product ID.

Clock Generator (CLK)

CLK_Output	Show how to clock output.
CLK_DeepSleep	Show how to entering and exiting deep sleep mode.
CLK_Sleep	Show how to entering and exiting sleep mode.

General Purpose I/O (GPIO)

GPIO_EINTAndNF	Show the usage of GPIO external interrupt function and noise filter function.
GPIO_OutputInput	Show how to set GPIO pin mode and use pin data input/output control.

Watchdog Timer (WDT)

WDT_ClearWindow	Demonstrate how to use WDT clear window function.
WDT_TimeoutAndReset	Demonstrate how to initialize WDT, reset WDT counter, and cause WDT error detection reset.

16-bit Timer (TIMER)

TIMER_Periodic	Use the Timer periodic mode to generate Timer interrupt every 0.05 second.
TIMER_PeriodicINT	Implement Timer counting in periodic mode.
TIMER_PWM_ChangeDuty	Change duty cycle and period of output waveform in PWM up count type.

TIMER_PWM_OutputWaveform	Demonstrate output different duty waveform in Timer0~3 PWM.
TIMER_Trigger	Start the timer with an external trigger.
TIMER_ToggleOut	Demonstrate the Timer toggle out function on Timer pin.
TIMER_EventCounter	Use Timer pin to demonstrate Timer event counter function.
TIMER_CaptureCounter	Show how to use the Timer capture function to capture Timer counter value.

Power Supply Voltage Detection (LVD)

LVD_BODWakeup	Demonstrate how to wake up system from Power-down mode by brown-out detector interrupt.

Clock Monitoring Circuit (CKM)

CKM_AfterPowerUp	After turning on the power of the MCU, the clock source of BASECLK is switched, and then the clock monitoring is started.
CKM_ChangingClock	The clock monitoring function is stopped, the clock monitoring frequency is changed, and the clock source of BASECLK is switched, and then the clock monitoring function is enabled.
CKM_AfterPowerUp_withClkTransitionMode	After turning on the power of the MCU, switch to the clock transition mode, set the clock monitoring frequency, switch the clock, and enable the clock monitoring function, and then cancel the clock transition mode.
CKM_ChangingClock_withClkTransitionMode	After switching to the clock transition mode, changing the clock monitoring frequency, and switching the BASECLK clock source, cancel the clock transition mode. By using the clock transition mode, the clock monitoring function stops only when BASECLK is switched to the built-in oscillation (HRCCLK), and the original BASECLK can always be monitored before and after switching.

Power Control PWM (GPWM)

GPWM_Brake	Demonstrate how to use GPWM brake function.
GPWM_DeadTime	Demonstrate how to use GPWM Dead Time function.
GPWM_DoubleBuffer	Change duty cycle and period of output waveform by GPWM Double Buffer function.
GPWM_OutputWaveform	Demonstrate how to use GPWM counter output waveform.
GPWM_SwitchDuty	Change duty cycle of output waveform by configured period.

Multi Feedback Assist (MFA)

MFA_Conflict	Demonstration of conflict detection function by MFA.
MFA_Blanking	Demonstration of blanking function by MFA.
MFA_Sync	Demonstration of synchronous start control function by MFA.

Digital-to-Analog Converter (DAC)

DAC_Output	Show how to DAC output.
DAC_SlopePWM	Demonstration of slope compensation by DAC.

Comparator (CMP)

CMP_CompareDAC	Demonstrate how to use comparator 2-level detection mode. (Reference voltage : DAC output)
CMP_CompareGPIO	Demonstrate how to use comparator 1-level detection mode. (Reference voltage : CMPREF input)
CMP_WindowCompare	Show how to monitor CMP input with window detection mode.

Analog to Digital Converter (ADC)

ADC_ADINT_Trigger	Use ADINT interrupt to do the ADC continuous scan conversion.
ADC_Multiple_Channel	Show how to convert multi-channel ADC.
ADC_Timer_Trigger	Show how to trigger ADC by Timer.
ADC_FaultCheck	Show how to use A/D fault check function.
ADC_TempSensor	Show how to use A/D Conversion of Temperature sensor.
ADC_Addition	Show how to use A/D conversion result addition function.

UART Interface Controller (UARTA)

UARTA_TxRxFunction	Demonstrate UARTA self loop back data transfer to show transmission and reception with one UART.
UARTA_DMA	Demonstrate UARTA self loop back data transfer with DMA.
UARTA_SingleWire	Demonstrate UARTA single wire data transfer.

UART Interface Controller (UARTB)

UARTB_TxRxFunction	Demonstrate UARTB self loop back data transfer to show transmission and reception with one UART.
UARTB_DMA	Demonstrate UARTB self loop back data transfer with DMA.
UARTB_SingleWire	Demonstrate UARTB single wire data transfer.

UART Interface Controller (UARTC)

UARTC_TxRxFunction	Demonstrate UARTC self loop back data transfer to show transmission and reception with one UART.
UARTC_DMA	Demonstrate UARTC self loop back data transfer with DMA.
UARTC_SingleWire	Demonstrate UARTC single wire data transfer.

Serial Peripheral Interface (SPIA)

SPIA_TxRxFunction	Demonstrate SPIA Master/Slave data transfer.
SPIA_DMA	Demonstrate SPIA Master/Slave data transfer with DMA.

Serial Peripheral Interface (SPIB)

SPIB_TxRxFunction	Demonstrate SPIB Master/Slave data transfer.
SPIB_DMA	Demonstrate SPIB Master/Slave data transfer with DMA.

Serial Peripheral Interface (SPIC)

SPIC_TxRxFunction	Demonstrate SPIC Master/Slave data transfer.
SPIC_DMA	Demonstrate SPIC Master/Slave data transfer with DMA.

I²C Serial Interface Controller (I2C)

I2C_GCMode_Slave	Show how a slave receives data from a master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
I2C_Slave	Demonstrate how to set I2C in Slave mode to receive 256 bytes data from a master. This sample code needs to work with I2C_Master.
I2C_GCMode_Master	Show how a master uses I2C address 0x0 to write data to a slave. This sample code needs to work with I2C_GCMode_Slave.
I2C_Master	Show how a master accesses a slave. This sample code needs to work with I2C_Slave.
I2C_MultiBytes_Master	Show how to set I2C Multi bytes API Read and Write data to Slave. This sample code needs to work with I2C_Slave.
I2C_SingleByte_Master	Show how to use I2C Single byte API Read and Write data to Slave. This sample code needs to work with I2C_Slave.
I2C_Wakeup_Slave	Show how to wake up MCU from Power-down mode via the I2C interface. This sample code needs to work with I2C_Master.

SMBus Interface (SMBUS)

SMBUS_GCMode_Slave	Show how a slave receives data from a master in GC (General Call) mode. This sample code needs to work with SMBUS_GCMode_Master.
SMBUS_Slave	Demonstrate how to set SMBus in Slave mode to receive 256 bytes data from a master. This sample code needs to work with SMBUS_Master.
SMBUS_GCMode_Master	Show how a master uses SMBus address 0x0 to write data to a slave. This sample code needs to work with SMBUS_GCMode_Slave.
SMBUS_Master	Show how a master accesses a slave. This sample code needs to work with SMBUS_Slave.
SMBUS_MultiBytes_Master	Show how to set SMBus Multi bytes API Read and Write data to Slave. This sample code needs to work with SMBUS_Slave.
SMBUS_SingleByte_Master	Show how to use SMBus Single byte API Read and Write data to Slave. This sample code needs to work with SMBUS_Slave.
SMBUS_Wakeup_Slave	Show how to wake up MCU from Power-down mode via the SMBus interface. This sample code needs to work with SMBUS_Master.

CRC Controller (CRC)

CRC_CCITT	Show how to CRC calculation function(CRC-16-CCITT)
CRC_CCITT_REV	Show how to CRC calculation function(CRC-16-CCITT_REV).

Flash Memory Controller (FMC)

FMC_DualBankFwUpdate	Implement a firmware update mechanism based on dual bank flash architecture.
FMC_ExeInSRAM	Erase and rewrite one sector of I-Flash by RAM execution.

DMA Controller (DMA)

DMA_BasicMode	Use DMA channel 1 to transfer data from memory to memory.
DMA_BurstRepeat	Demonstrate DMA burst repeat mode to transfer data from memory to memory.

CAN Controller (CAN)

CAN_NormalMode_Rx	Demonstrate CAN bus receive a message with normal mode.
CAN_NormalMode_Tx	Demonstrate CAN bus transmit a message with normal mode.

Real Time Clock (RTC)

RTC_Alarm_Test	Demonstrate the RTC alarm function.
RTC_Alarm_Wakeup	Use RTC alarm interrupt event to wake up system.
RTC_Time_Display	Demonstrate the RTC function and displays current time to the UART console.

LIN Controller (LIN)

LIN_Master	Show how to use LIN communication as a master node. This sample code needs to work with LIN_Slave.
LIN_Slave	Show how to use LIN communication as a slave node. This sample code needs to work with LIN_Master.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.